RSRE MEMORANDUM No. 4358

ROYAL SIGNALS & RADAR ESTABLISHMENT

IMPROVED FRONT-END ANALYSIS IN THE *ARM* SYSTEM: LINEAR TRANSFORMATIONS OF SRUbank

Authors: M J Russell, D Lowe, M D Bedworth & K M Ponting

PROCUREMENT EXECUTIVE,
MINISTRY OF DEFENCE,
R S R E MALVERN,
WORCS.

RSRE MEMORANDUM No. 4358

UNLIMITED

Royal Signals and Radar Establishment Memorandum 4358

Improved Front-End Analysis in the *ARM* System: Linear Transformations of SRUbank

M J Russell, D Lowe, M D Bedworth and K M Ponting
Speech Research Unit, SP4,
Royal Signals and Radar Establishment,
St. Andrews, Great Malvern, England

13th February 1990

Abstract

Front-end acoustic analysis in early versions of the Airborne Reconnaissance Mission (*ARM*) continuous speech recognition system was based on the SRUbank filterbank analyser. In its default configuration, this is a conventional, high-resolution filterbank analyser with 27 critical band filters spanning the range 0 to 10kHz and producing 100 frames per second. This memorandum reports experiments which show that recognition accuracy is improved by applying a suitable dimension-reducing linear transformation to the output of SRUbank. Experiments were conducted using several linear transformations of SRUbank, including 8, 12 and 16 cosine coefficients plus mean channel amplitude, 8, 12 and 16 cosine coefficients plus mean channel amplitude plus difference between corresponding elements of the feature vector at ±20 milliseconds, and 8 and 16 principal components.

1 Introduction

The work described in this memorandum was conducted at the UK Speech Research Unit as part of the Airborne Reconnaissance Mission (*ARM*) continuous speech recognition project. The aim of the *ARM* project is accurate recognition of continuously spoken airborne reconnaissance reports using a speech recognition system based on phoneme-level hidden Markov models (HMMs). The *ARM* project is described in [2]. The work described here uses *ARM-3(IF)*, an early version of the *ARM* system.

Front-end acoustic analysis in *ARM-3(IF)* is based on a conventional filter-bank analyser with 27 critical band filters spanning the range 0 to 10kHz. However, it has been shown elsewhere (e.g.[4]) that the recognition accuracy of such a system is improved if this type of spectral representation is replaced with a cepstral representation, and if information about spectral change with respect to time is made available to the recogniser.

In the context of the *ARM* system a cepstral representation is obtained by transforming the SRUbank output vectors using a discrete cosine transform [3]. For typical speech signals this has the effects of diagonalising the covariance matrix of the data (i.e. removing correlation between the different cosine coefficients) and concentrating the variance of the data into the lower dimensions. This raises two issues. First, since the higher cosine coefficients carry virtually no variance it is often assumed that they are at best irrelevant to the classification process and at worst act as noise. (In general this assumption is false, since variance is not a universal guide to discrimination power. It is easy to construct simple two-class examples where all discriminative information is carried by the direction of minimum overall variance and none by the direction of overall maximum variance). Therefore the question of how many, and which, cosine coefficients should be used arises. Second, if any performance improvement which results from the application of the cosine transform is due to the diagonalisation of the covariance matrix, then one would expect to observe a larger improvement if the cosine transform were replaced by the linear transformation which results from a principal components analysis, since the latter is specifically designed to diagonalise the covariance matrix. Both of these issues are addressed by the experiments reported in this memorandum.

Information about how the cepstrum changes with respect to time can be included in the front end representation by augmenting the feature vector at time t with the difference between feature vectors at times $t \pm \delta$ for some $\delta > 0$. Since this has the effect of doubling the dimensionality of the acoustic vectors (and hence doubling the number of system parameters) there is a danger that any improvement in the quality of the front-end might be overshadowed by undertraining due to an inability of the training data to support the increased number of parameters. Since all HMMs in *ARM-3(IF)* have Gaussian state output pdfs with diagonal covariance

matrices, a solution to this problem is to estimate a common shared, or grand, covariance matrix for all states of the HMMs in the model set [5]. Despite its simplicity, this method has been shown to be extremely effective in improving the performance of systems with a large number of parameters [5, 2]

This note reports the results of experiments which compare the recognition accuracy obtained using front-end representations consisting of 8, 12 and 16 cosine coefficients plus mean channel amplitude, and 8 and 16 principal components. Experiments are also reported in which 8, 12 and 16 cosine coefficients plus mean channel amplitude are supplemented with the differences between correponding elements of the feature vectors at ± 20 milliseconds. The latter experiments use both state specific and grand covariance matrices in the HMM state output probability density functions.

2 *ARM* Version 3*IF*

Front-end acoustic analysis in *ARM-3(IF)* is based on the SRUbank filterbank analyser with default settings (27 critical band filters spanning the range 0 to 10kHz, 100 frames per second). The feature vector o_t at time t is obtained from the SRUbank output vector v_t by subtracting the mean channel amplitude $m(v_t)$ from each component of v_t and setting the 28^{th} component of o_t equal to $m(v_t)$. More precisely,

$$o_t{}^d = v_t{}^d - m(v_t),\ d = 1, ..., 27$$
$$o_t{}^{28} = m(v_t)$$

where

$$m(v_t) = \frac{1}{27}\sum_{d=1}^{27} v_t{}^d$$

Acoustic-phonetic processing in *ARM-3(IF)* uses a set of 72 HMMs consisting of:

- Sixty-two three-state phoneme-level HMMs, comprising one HMM for each vowel plus separate HMMs for all syllable initial and syllable final consonants which occur in the *ARM* vocabulary.
- Four single state "non-speech" HMMs to cope with non-speech sounds in regions of the test data between spoken sentences.
- Six word-level HMMs for the commonly occuring short words "air", "at", "in", "of", "oh" and "or". The number of states in these word-level models is equal to three times the number of phonemes in their baseform transcriptions.

All HMM states in *ARM-3(IF)* are identified with single multivariate Gaussian state output probability density functions with diagonal (co)variance matrices.

3 The Discrete Cosine Transform

The discrete cosine transform used in the experiments is defined as follows. As above, each 27 dimensional SRUbank output vector $v = (v^1, v^2, ..., v^{27})$ was first amplitude normalised to produce a new vector w, given by:

$$w = v - m(v).i^*$$

Where $i = (1, 1, ..., 1)$. The discrete cosine transform C is defined by

$$C(w)^j = \sqrt{\frac{2}{27}} \sum_{d=1}^{27} w^d cos(\frac{\pi \times j \times (d - 0.5)}{27})$$

for $j = 1$ to 26.

Figure 1 shows a graph of variance as a function of cosine coefficient computed over approximately 56 seconds of transformed speech. The figure confirms that most of the variance is carried in the lower cosine coefficients, with a large drop in variance at the 8^{th} coefficient and very little variance beyond the 16^{th} . Based on this data it was decided to conduct experiments using front-end representations consisting of 8, 12 and 16 cosine coefficients plus mean SRUbank channel amplitude. In what follows these three representations will be refered to as *CC8*, *CC12* and *CC16* respectively. Notice that *CCd* is a $d + 1$ dimensional representation with the mean SRUbank channel amplitude forming the $d + 1^{th}$ component. The latter replaces the zeroth cosine coefficient (which is a linear function of the channel mean), which is not used. FORTRAN code for computing this representation from the original SRUbank data is included in appendix A.

4 The Discrete Cosine Transform plus Time Differences

It has been shown elsewhere (e.g. [4]) that recognition accuracy is improved if information concerning the change in feature vectors with respect to time is included in the front end representation. Accordingly three additional front-end representations *CC8* δ, *CC12* δ and CC16 δ were included in the experiments. These representations are defined as follows. If u_t denotes the vector at time t for representation *CCd* then the corresponding vector u'_t for representation *CCd* δ is the $2d+2$ dimensional vector defined by:

$u_t'^d = u_t{}^d$, for $d = 1, ..., n + 1$

$u_t'^d = u_{t+\delta}{}^d - u_{t-\delta}{}^d$, for $d = n + 2, ..., 2n + 2$.

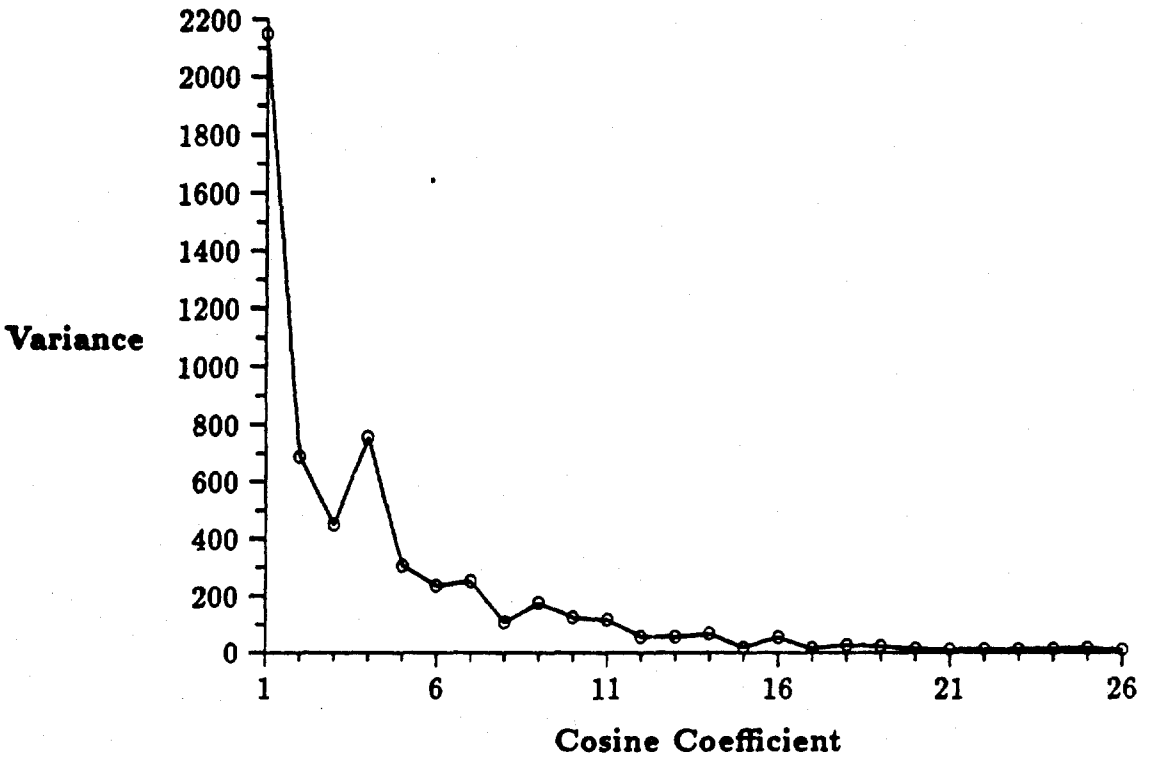

Figure 1: *Variance as a function of cosine coefficient*

In the present experiments $\delta = 2$. Notice that these representations include mean SRUbank channel amplitude at time t (component $d + 1$) and difference between mean channel amplitudes at times $t \pm \delta$ (component $2d + 2$).

5 Grand Variance

Experiments were conducted using the *CC8 δ* and *CC16 δ* front-ends in which all of the states in the HMMs in the model set shared a common, grand diagonal (co)variance matrix [5]. These parametrisations will be refered to as *CC8 δ GV* and *CC16 δ GV* repectively.

For information, figure 2 shows grand variance as a function of cosine coefficient for the *CC16* representation, computed during HMM parameter reestimation. The smoother form of this graph compared with figure 1 follows from the fact that grand variance is computed over sets of feature vectors which are mapped onto the same state by the forward-backward algorithm during the reestimation process rather than over all feature vectors. The increase in grand variance for the 17^{th} component occurs because this component corresponds to mean channel amplitude.

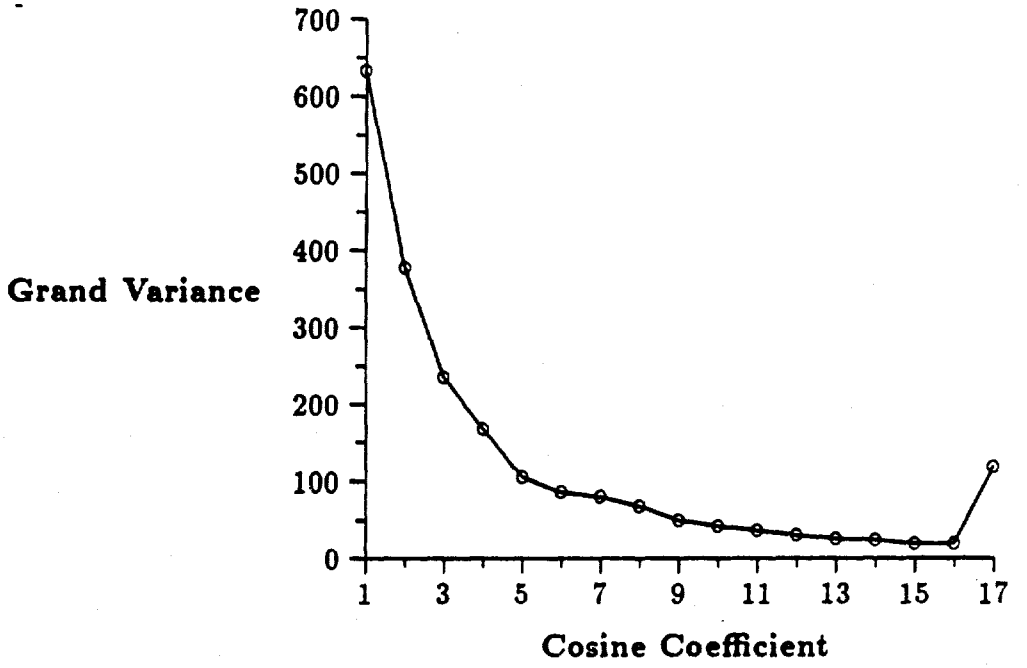

Figure 2: *Grand variance as a function of cosine coefficient*

6 Principal Component Analysis.

The Cosine Transform is one example of an unsupervised feature extraction mechanism which is also dimension-reducing. This section describes another linear feature extraction mechanism, *Principal Component Analysis*. This technique is equivalent to finding a rotation of the original pattern space (the 27 dimensional SRUbank space in this case) to a new set of orthogonal vectors. The rotation is such that the projection of the data onto a subset of the new orthogonal basis vectors is optimum from the point of view of data reconstruction. The mathematics summarising this process follows.

Consider a set of P patterns in an n dimensional Euclidean space $\{\boldsymbol{x}_p \in \mathbb{R}^n \quad p = 1,2,\ldots P\}$. It is convenient to work with zero mean data, i.e. $\boldsymbol{x}_p \rightarrow \boldsymbol{z}_p = \boldsymbol{x}_p - \overline{\boldsymbol{x}}$ where $\overline{\boldsymbol{x}}$ is the sample mean of the data set. Of course, one may express $\boldsymbol{z}$ in terms of the projections onto any set of n orthogonal basis vectors, $\{\boldsymbol{e}_i \in \mathbb{R}^n \quad i = 1,2,\ldots n\}$. However, we are more interested in the projection onto a finite subset, say r, of the orthogonal vectors. The criterion for choosing these basis vectors is to minimise the sum square error between the actual set of patterns and the reconstructed set of patterns. Thus we wish to minimise

$$E_{(r)} = \sum_{p=1}^{P} \| \boldsymbol{z}_p - \sum_{k=1}^{r} a_{pk}\boldsymbol{e}_k \|^2 \qquad (1)$$

Due to the assumed orthonormality of $\{\boldsymbol{e}_i \; i = 1,2,\ldots n\}$, the expansion coefficients

are given explicitly as

$$a_{pk} = x_p^* . e_k \tag{2}$$

Substituting (2) in (1) and rearranging gives the error to be minimised as

$$\begin{aligned} E &= \sum_{p=1}^{P} \| \sum_{k=r+1}^{n} a_{pk} e_k \|^2 \\ &= \sum_{p=1}^{P} \| \sum_{k=r+1}^{n} (x_p^* . e_k) e_k \|^2 \\ &= \sum_{p=1}^{P} \sum_{k=r+1}^{n} e_k^* . x_p x_p^* . e_k \\ &= \sum_{k=r+1}^{n} e_k^* . \Phi . e_k \end{aligned} \tag{3}$$

where Φ is the covariance matrix of the original data. It can be shown (by the method of Lagrange multipliers) that the minimum of the above error expression is obtained when the orthogonal basis vectors, e_k are chosen to be the eigenvectors of the data covariance matrix,

$$\Phi e_k = \lambda_k e_k \tag{4}$$

Thus, using (4) in (3) shows that the minimum estimated reconstruction error obtainable is given by

$$E_r = \sum_{k=r+1}^{n} \lambda_k \tag{5}$$

Therefore, E_r will be a minimum if the first r eigenvectors of the covariance matrix corresponding to the r largest eigenvalues are chosen.

This illustrates why the optimum projection from the point of view of reconstruction error is onto the most significant eigenvectors of the covariance matrix, if one wishes to choose a linear transformation of data followed by a low dimensional projection. One should bear in mind that if the aim is *not* to obtain minimum reconstruction error (for instance in a class discrimination experiment) the principal component analysis may not be good transformation to employ. Although one might hope intuitively that a projection onto a subspace which retains the maximum variance in the data also preserves that information which is most vital for discriminating between classes, this is not guaranteed. Indeed, the experimental results in this memorandum support this warning.

6.1 Details of Implementation

For the analysis of the 27 dimensional SRUbank data, it is necessary to find the eigenvectors of the covariance matrix of a large data set. Rather than evaluate

the covariance matrix (which would introduce numerical uncertainties due to the large numbers involved in summing many thousands of vectors), the eigenvectors may be obtained by a singular value decomposition of the data matrix itself. Any (rectangular) matrix, **A** of size $m \times n$ ($m > n$) may be decomposed into the product of three other matrices as

$$\mathbf{A} = \mathbf{UQV}^* \tag{6}$$

where **U** is an $m \times n$ matrix whose colums are orthogonal, **Q** is a diagonal $n \times n$ matrix whose elements (the *singular values*) are the positive square roots of the eigenvalues of the covariance matrix $\mathbf{A}^*\mathbf{A}$ and **V** is an $n \times n$ matrix whose columns are the eigenvectors of the covariance matrix. This is known as a singular value decomposition and there are efficient algorithms for its numerical implementation. We chose to obtain the covariance matrix eigenvectors by using this singular value decomposition technique. This has the advantage that the distribution of the singular values usually conveys knowledge as to the distribution of information in the data – one would expect that the exponentially decaying tail of the singular values is dominated by noise (either intrinsic to the data, or noise due to numerical precision). Also in real data in practice one can often discern 'discontinuities' in the graph of singular values plotted as a function of order. These discontinuities are indicative of a significant change in variance of the projected data and consequently are good indicators to decide the order, or dimensionality of the projected data.

The amount of data to use was decided by using a sufficient number of frames such that the successive ratios of the singular values of the data matrix did not change by adding more data. Note that the magnitude of the singular values increases as the amount of data increases, but the ratio of the $(n+1)$th to the nth singular value should remain fixed if there is sufficient data for reliable estimation. For the experiment, 30000 frames of SRUbank data were used from the training set described in section 7.1. This corresponded to the first 266 utterances in the training set.

A singular value decomposition of the data matrix was performed which produced the required eigenvectors and singular values. Once it was decided how many singular values were important, the original 27 dimensional pattern vectors for *all* the files of interest were projected onto the subspace spanned by those significant singular vectors. The transformed vector components were obtained according to equation (2). The experimentally observed singular values are plotted in figure 3 and the eigenvectors corresponding to the largest five singular values are displayed in figure 4. The original 27 dimensional SRUbank data from all files were projected onto the first 8 and 16 singular vectors. The resulting parameterisations will be refered to as *PC8* and *PC16* respectively. These reduced dimension patterns were used in the same manner as the cosine transform vectors in the HMM recognition experiment.

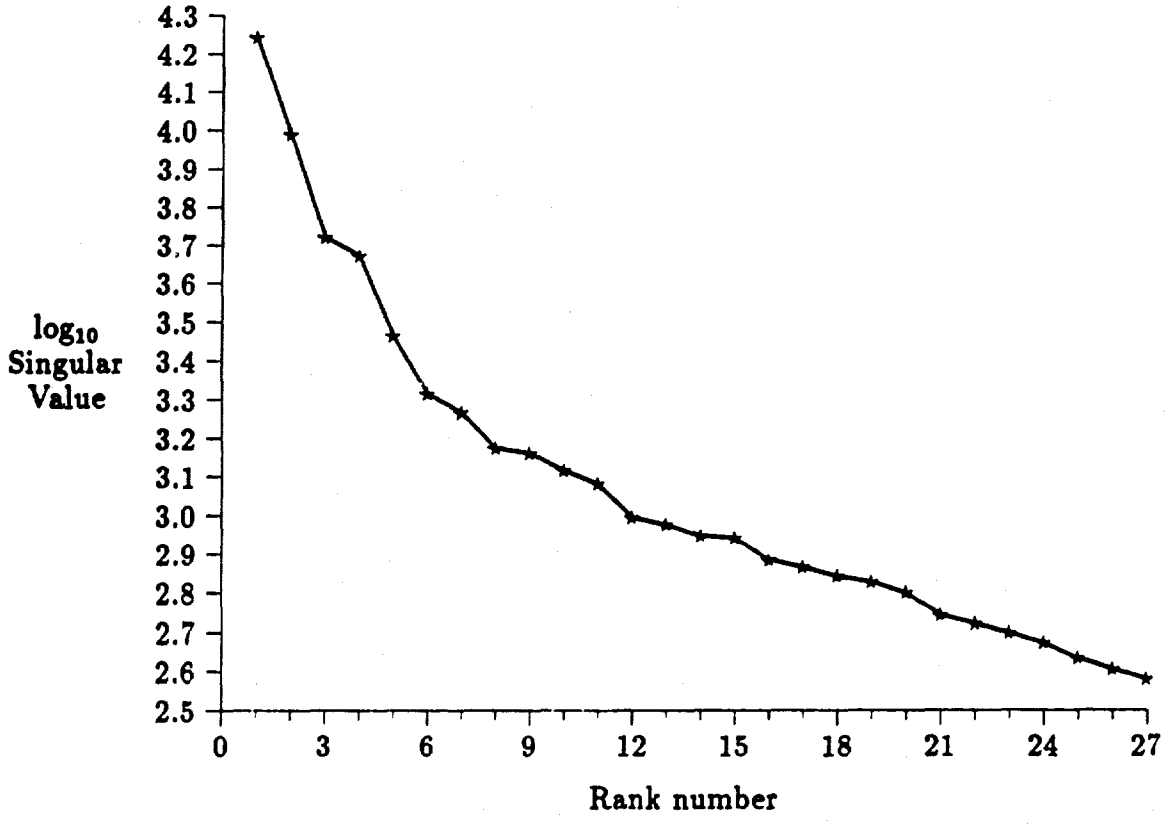

Figure 3: The singular values of SRUbank data derived from speaker SJ

7 Experimental Method

7.1 Training and Test Data

The experiments were conducted using speech from a single speaker (SJ) as training and test material. The training set consisted of 37 *ARM* reports (224 sentences, 1985 words) chosen to give maximum coverage of phonemes which occur infrequently in the *ARM* vocabulary. Ten reports from the same speaker (540 words, 2293 phonemes (according to baseform transcriptions)) were used as test material. This size of test set was identified as the minimum necessary to ensure statistical significance of the improvements in word accuracy (5% improvements up to an absolute word accuracy of 70%) which were expected during the initial development of the *ARM* system.

7.2 HMM Training

Initial estimates of the parameters of the phoneme HMMs were obtained from the equivalent of two *ARM* reports of speech, hand labelled at the phoneme level. Similarly, initial estimates of the common word HMM parameters were obtained from

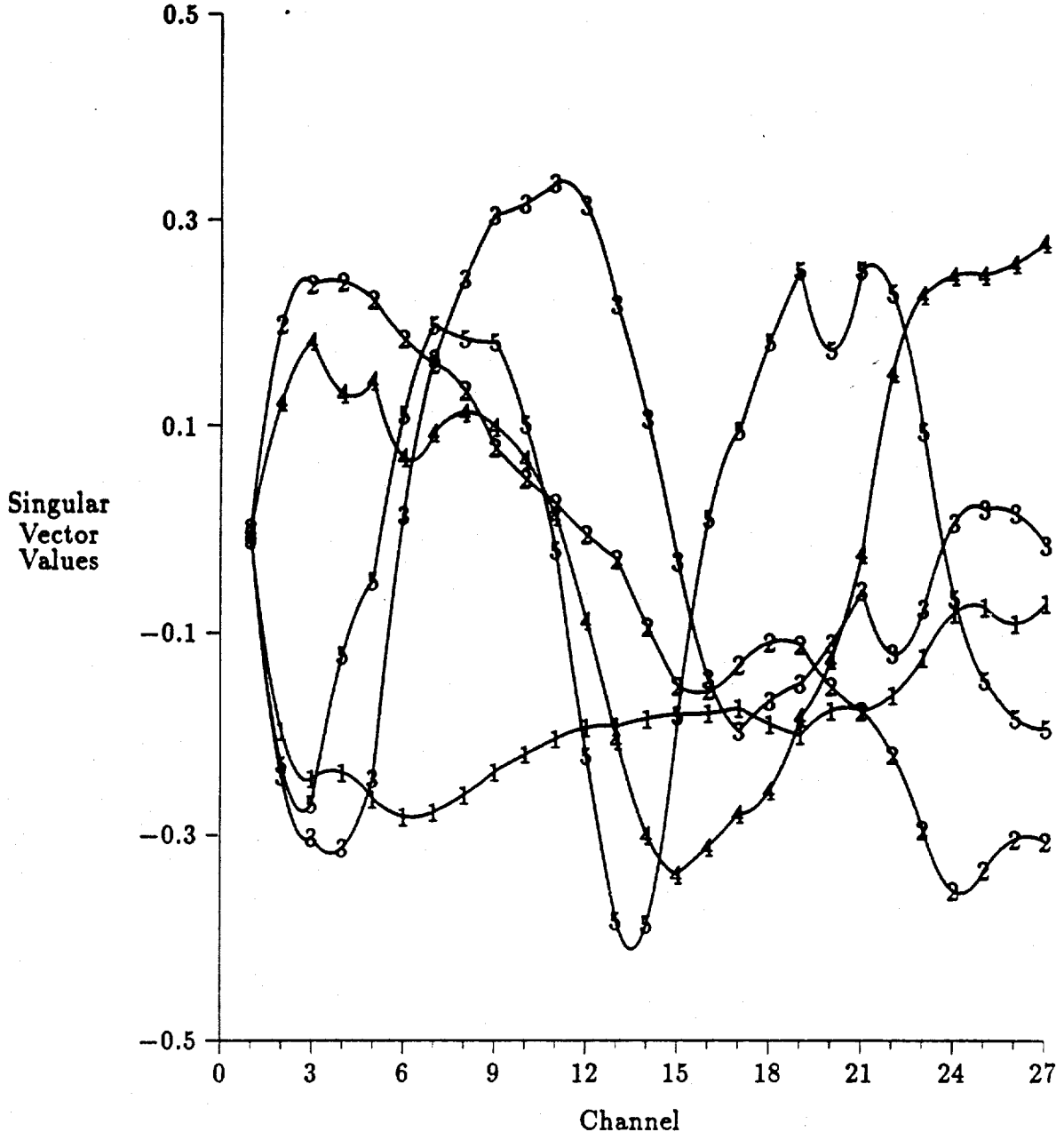

Figure 4: The eigenvectors corresponding to the five largest singular values

single examples of these words extracted from continuous speech. The initial estimates of parameters of a single state "non-speech" HMM were obtained from a typical non-speech region of the training data. This model was used as the initial model for all four non-speech HMMs. The models were optimised with respect to the complete training set labelled orthographically at the sentence level. Standard sub-word HMM training procedures were used in which sentence level HMMs were constructed from phoneme-level HMMs using a dictionary of baseform transcriptions of *ARM* vocabulary words. These models were then mapped onto the sentence level acoustic data using the forward backward algorithm to obtain contributions to the model parameter estimates.

7.3 Recognition

Recognition was performed using a one-pass dynamic programming algorithm with beam search and partial traceback [1]. Results are presented in terms of *% words correct* and *% word accuracy*. These are computed as follows, using dynamic programming to align the true transcription of the test data with the output of the recogniser:

$$\%\ words\ correct = \frac{N - S - D}{N} \times 100,$$

$$\%\ word\ accuracy = \frac{N - S - D - I}{N} \times 100$$

where N is the number of words in the test set, and S, D and I are the number of words recognised as the incorrect word, deleted and inserted respectively.

Three different syntaxes were used to constrain the recognition process: a *word* syntax, which allows recognition of any sequence of words from the ARM vocabulary; a *full* syntax (perplexity 6) which was used to generate the ARM reports, and a phoneme based *simple* syntax which allows any sequence of phonemes to be recognised.

8 Results

The results of all of the experiments are presented in table 1.

Concentrating first on the figures for word accuracy, the table shows that the *CC12*, *CC16*, *CC12 δ* and *PC16* representations all perform significantly better than the original SRUbank-based front-end, and that the best performance is obtained using the *CC12* and *CC16* representations. The *CC8* and *CC8 δ* representations perform slightly (but not significantly) worse than SRUbank. Significantly poorer performance results when grand variance is used in conjunction with the *CC8 δ* and

CC16 δ representations, suggesting that undertraining is not a problem with this version of the *ARM* system.

The figures for phoneme accuracy lead to a different interpretation of the results. In this case best performance is obtained using the representations *CC8 δ*, *CC12 δ* and *CC16 δ* which include information about temporal dynamics. The figure for *CC16 δ* is particularly interesting since this representation leads to a significant increase in phoneme accuracy but decrease in word accuracy with respect to SRUbank. This suggests that the *CC16 δ* front-end changes the distribution of phoneme errors so as to increase overall phoneme accuracy but at the same time reduce the recognition performance for some phonemes which are important for word- level recognition. This phenomenon has not been further investigated.

The two *PC* parameterisations perform worse than their *CC* counterparts in terms of both phoneme and word recognition accuracy. One suggestion for this unexpected result is that the superior performance obtained with the cosine transform is due to more than the cosine transform's ability to remove correlation between the different components of the feature vectors. An alternative possibility is that the better performance of the *CC* paramterisations is due to the explicit inclusion of the original SRUbank mean channel amplitude.

The best overall performance is obtained with the *CC12 δ* parameterisation.

	Phoneme Syntax (Perplexity=47)		Word Syntax (Perplexity=497)		Full Syntax (Perplexity=6)	
Front-end	Phonemes Correct	Phoneme Accuracy	Words Correct	Word Accuracy	Words Correct	Word Accuracy
SRUbank	59.4%	40.9%	80.6%	56.9%	98.7%	98.5%
CC8	56.0%	38.9%	78.7%	53.3%	98.1%	97.4%
CC12	61.4%	44.3%	84.3%	63.5%	98.0%	97.0%
CC16	62.7%	44.6%	83.7%	62.2%	98.5%	97.8%
CC8 δ	65.8%	49.4%	82.2%	56.1%	98.3%	97.8%
CC12 δ	67.9%	54.1%	85.0%	62.0%	98.5%	97.6%
CC16 δ	66.9%	49.4%	80.4%	52.2%	98.5%	97.8%
PC8	45.7%	35.8%	78.7%	45.6%	96.7%	95.4%
PC16	58.9%	42.3%	83.3%	60.3%	97.8%	96.1%
CC8 δ GV	58.0%	39.3%	76.3%	34.1%	98.9%	98.7%
CC16 δ GV	60.7%	39.8%	77.4%	35.4%	98.0%	97.0%

Table 1: Results of experiments using alternative linear transformations of the SRUbank front-end (speaker SJ, 540 word test set).

9 Conclusions

The experiments reported in this memorandum show that the performance of the *ARM* system can be significantly improved by applying a suitable linear transformation to the output of the SRUbank filterbank analyser.

Subsequent work on the *ARM* system has been conducted using the *CC16* and *CC12 δ* parametrisations.

References

[1] J S Bridle, M D Brown and R M Chamberlain, "A one-pass algorithm for connected word recognition", IEEE-ICASSP, 899-902, 1982.

[2] M J Russell, K M Ponting, S M Peeling, S R Browning, J S Bridle and R K Moore, "The ARM Continuous Speech Recognition System", Proc. ICASSP'90, Albuquerque, New Mexico, April 1990.

[3] J N Holmes, "Speech Synthesis and Recognition", Van Nostrand Reinhold (UK), 1988.

[4] K-F Lee, "Large Vocabulary Speaker-Independent Continuous Speech Recognition: the SPHINX System", PhD Thesis, Carnegie Mellon University, 1988.

[5] D B Paul, "A speaker-stress resistant isolated word recognizer", ICASSP'87, Dallas, TX, 1987.

Appendix A

FORTRAN source code used to compute cosine transform based front-end representation. The motivation for including this in the text is to define unambiguously the implementation of the cosine transform which was used. The code, as presented, is inefficient due to the computation of the cosine term in the inner-most loop.

```
C+
C ** COMPUTE MFCC SCALE CONSTANT
C-
const=SQRT(2/srubank_dim)

DO t=1,num_frames
C+
C  ** AMPLITUDE NORMALISE
C-
 mean_channel_amp=0
 DO i=1,srubank_dim
  mean_channel_amp=mean_channel_amp+srubank_frame(i,t)
 END DO
 mean_channel_amp=mean_channel_amp/srubank_dim
 DO i=1,srubank_dim
  srubank_frame(i,t)=srubank_frame(i,t)-mean_channel_amp
 END DO
C+
C  ** MFCC CALCULATION
C-
 DO i=1,num_cosine_coeffs
  cc_frame(i,t)=0
  DO j=1,srubank_dim
   cos_term=COS(i*(j-0.5)*PI/srubank_dim)
   cc_frame(i,t)=cc_frame(i,t)+ srubank_frame(j,t)*cos_term
  END DO
  cc_frame(i,t)=const*cc_frame(i,t)
 END DO
 cc_frame(num_cosine_coeffs+1)=mean_channel_amp

END DO ! end of t loop
```

REPORT DOCUMENTATION PAGE

DRIC Reference Number (if known) .....

Overall security classification of sheet .....Unclassified.....

(As far as possible this sheet should contain only unclassified information. If it is necessary to enter classified information, the field concerned must be marked to indicate the classification eg (R), (C) or (S).

Originators Reference/Report No.	Month	Year
MEMO 4358	FEBRUARY	1990

Originators Name and Location

RSRE, St Andrews Road
Malvern, Worcs WR14 3PS

Monitoring Agency Name and Location

Title

IMPROVED FRONT-END ANALYSIS IN THE *ARM* SYSTEM:
LINEAR TRANSFORMATIONS OF SRUbank

Report Security Classification	Title Classification (U, R, C or S)
Unclassified	U

Foreign Language Title (in the case of translations)

Conference Details

Agency Reference	Contract Number and Period

Project Number	Other References

Authors	Pagination and Ref
Russell, M J; Lowe, D; Bedworth, M D; Ponting, K M	14

Abstract

Front-end acoustic analysis in early versions of the Airborne Reconnaissance Mission (*ARM*) continuous speech recognition system was based on the SRUbank filterbank analyser. In its default configuration, this is a conventional, high-resolution filterbank analyser with 27 critical band filters spanning the range 0 to 10 kHz and producing 100 frames per second. This memorandum reports experiments which show that recognition accuracy is improved by applying a suitable dimension-reducing linear transformation to the output of SRUbank. Experiments were conducted using several linear transformations of SRUbank, including 8, 12 and 16 cosine coefficients plus mean channel amplitude, 8, 12 and 16 cosine coefficients plus mean channel amplitude plus difference between corresponding elements of the feature vector at ±20 milliseconds, and 8 and 16 principal components.

Abstract Classification (U,R,C or S)
U

Descriptors

Distribution Statement (Enter any limitations on the distribution of the document)

Unlimited

S80/48